COURSE CONTENT

ENGINEERING DEVELOPMENT PROJECT								
VI Semester: COMMON TO ALL BRANCHES								
Course Code	Category	Hours/Week			Credits	Maximum Marks		
ACSD45	SKILL	L	T	P	C	CIA	SEE	Total
		0	0	2	1	40	60	100
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				Total Classes: 32		
Prerequisite: Theory of Structures								

I. COURSE OVERVIEW:

The Engineering Development Project is a practice-oriented course designed to provide students with hands-on experience in applying engineering principles, innovation frameworks, and problem-solving skills to real-world challenges. This course enables students to conceptualize, design, develop, test, and deliver a complete engineering solution through a structured project lifecycle. It begins with project initiation and planning, where students identify relevant problems through brainstorming and literature review, define project objectives, assess feasibility, and prepare a formal project proposal. Emphasis is placed on aligning projects with societal needs, sustainability goals, and emerging technologies. In the design and development phase, learners create system architectures, module designs, and interfaces, followed by implementation of core functionalities. It strengthens problem-solving, teamwork, and innovation skills while preparing students for real-world applications and professional practice.

II. COURSE OBJECTIVES:

The students will try to learn:

i. To enable students to identify real-world problems and translate them into feasible engineering projects through systematic problem definition, requirement analysis, and planning.

ii. To develop students' ability to design, develop, and implement engineering solutions, including system architecture, module integration, testing, and refinement uses appropriate tools and technologies.

iii. To promote innovation, teamwork, and project management skills by guiding students through structured project phases, reviews, documentation, and presentations aligned with TRL and YUKTI guidelines.

iv. To enhance professional competencies such as communication, critical thinking, and technical documentation through project reports, demonstrations, viva voce, and evaluation by expert panels.

III. COURSE OUTCOMES:

At the end of the course students should be able to:

CO1 Understand and explain the fundamentals of engineering project development, including problem identification, project planning, feasibility analysis, and Technology Readiness Levels (TRLs).

CO2 Apply systematic methods to identify real-world problems, define project requirements, conduct literature surveys, and prepare a structured project proposal.

CO3 Analyze system requirements and design solutions, including system architecture, module design, interface specifications, and selection of appropriate tools and technologies.

CO4 Develop and integrate functional project modules, implement coding solutions, and perform unit-level testing to validate system performance.

CO5 Evaluate project outcomes through system testing, debugging, refinement, and incorporation of feedback based on predefined technical and performance criteria

CO6 Create a complete engineering solution, including final project implementation, documentation, presentation, and viva voce, demonstrating innovation, teamwork, and professional competence.

IV. COURSE CONTENT:

ENGINEERING DEVELOPMENT PROJECT

1.1 SECTION 12: Course Content	
WEEK- I	**Orientation to project course, Team formation & role assignment, Brainstorming project ideas, Initial literature survey.**
	1. Understand the structure, objectives, and expectations of the Engineering Development Project course, including deliverables, evaluation criteria, and alignment with YUKTI innovation guidelines. 2. Form effective multidisciplinary teams by identifying individual strengths and assigning appropriate roles such as team leader, designer, developer, researcher, and documentation lead. 3. Apply brainstorming and ideation techniques to generate innovative, feasible, and problem-oriented project ideas addressing real-world challenges and SDGs. 4. Identify and articulate a relevant engineering problem through discussion, observation, and preliminary analysis. 5. Conduct an initial literature survey using research papers, patents, case studies, and existing solutions to understand the state-of-the-art and research gaps. 6. Differentiate between existing solutions and proposed ideas, establishing novelty, innovation potential, and scope for development. 7. Develop collaborative and communication skills through teamwork, discussion, and decision-making during the early project stages.
WEEK- 2	**Problem identification & requirement gathering, Define scope, objectives, and deliverables, Draft project proposal.**
	1. Identify and analyze real-world engineering problems by engaging with societal, industrial, or community needs through systematic observation and research. 2. Elicit, document, and prioritize functional and non-functional requirements using appropriate requirement-gathering techniques such as stakeholder interviews, surveys, and benchmarking. 3. Define clear project scope, objectives, and deliverables by translating problem statements into measurable and achievable engineering goals. 4. Develop a structured project proposal that includes problem definition, objectives, scope, methodology, timelines, and expected outcomes in a professional format. 5. Evaluate the feasibility and relevance of the proposed project idea in terms of technical, social, economic, and sustainability aspects, including alignment with SDGs/YUKTI guidelines. 6. Demonstrate professional communication and documentation skills through well-written proposals and effective presentation of project intent.
WEEK- 3	**Feasibility study (technical, economic, social), Tools & technology selection, Project plan with milestones.**
	1. Analyze technical feasibility of proposed engineering solutions by evaluating available technologies, resources, and implementation constraints. 2. Evaluate economic and social feasibility by estimating costs, benefits, risks, and societal impact of the proposed project. 3. Select appropriate tools, technologies, and platforms based on project requirements, innovation potential, and alignment with YUKTI/TRL guidelines. 4. Develop a structured project plan with milestones, timelines, and deliverables using standard project management techniques. 5. Justify technology and planning decisions through feasibility findings and documented reasoning. 6. Demonstrate readiness for project execution by integrating feasibility outcomes into a realistic and scalable implementation roadmap.

WEEK- 4	**System Architecture Design (UML diagrams, flowcharts), Database Schema Design**
	1. Understand the principles of system architecture and how to structure an engineering solution into logical components and subsystems. 2. Develop the ability to represent system workflows and interactions using UML diagrams (use case, class, sequence) and flowcharts. 3. Apply architectural design concepts to translate problem requirements into structured technical designs. 4. Design an appropriate database schema (if applicable) to ensure efficient data storage, retrieval, and integrity. 5. Analyze relationships between system components, data flow, and user interactions to support scalable and maintainable solutions. 6. Demonstrate the ability to document system architecture clearly to support implementation, testing, and future enhancements.
WEEK- 5	**Module design & interface specification, Prepare test plan**
	1. Design modular system components by decomposing the overall system architecture into well-defined functional modules with clear responsibilities. 2. Develop interface specifications that define inputs, outputs, data flow, and interactions between different software/hardware modules. 3. Apply design principles such as modularity, cohesion, and low coupling to improve system scalability, maintainability, and reliability. 4. Prepare a comprehensive test plan outlining testing objectives, test cases, test data, and expected outcomes for validating module functionality
WEEK- 6	**Begin coding core modules, Weekly progress review**
	1. Apply appropriate programming languages, frameworks, and tools to implement core functional modules of the proposed engineering solution. 2. Translate design specifications and system architecture into executable code with proper logic and structure. 3. Practice version control, coding standards, and documentation for maintainable and scalable development. 4. Monitor project progress through weekly reviews, identifying technical challenges and corrective actions.
WEEK- 7	**Continue development, Integration of modules, Mid-term presentation & evaluation**
	1. Apply systematic development practices to continue building and refining individual project modules based on design specifications. 2. Integrate multiple software/hardware modules into a cohesive system, ensuring compatibility, data flow, and functional consistency. 3. Analyze integration issues and resolve functional, logical, or performance-related conflicts during system assembly. 4. Demonstrate project progress through a structured mid-term presentation, clearly articulating design choices, development status, and challenges. 5. Evaluate their project outcomes against predefined objectives and rubrics through mid-term assessment and expert feedback. 6. Incorporate constructive feedback from the evaluation to improve system functionality, design quality, and project direction.
WEEK- 8	**Unit testing of modules, Debugging & refinement**
	1. Understand the importance of unit testing in ensuring the correctness and reliability of individual software/hardware modules.

	2. Apply appropriate unit testing techniques and tools to verify the functionality of developed modules. 3. Analyze test results to identify logical
WEEK- 9	**System testing (functional, performance, security), Documentation of test results**
	1. Understand System Testing Concepts 2. Perform Functional Testing 3. Evaluate Performance of the System. 4. Assess System Security 5. Analyze Test Results 6. Document Test Activities and Results
WEEK- 10	**User interface polishing, Incorporate feedback from mid-term review**
	1. Improve the visual appearance, consistency, and usability of the user interface based on design principles and usability standards. 2. Apply User-Centered Design Principles 3. Analyze Mid-Term Review Feedback 4. Improve System Usability and Satisfaction 5. Verify that UI changes align with project requirements and do not negatively impact existing functionality. 6. Present a polished, user-friendly interface that reflects responsiveness to feedback and readiness for final assessment or deployment.
WEEK- 11	**Prepare draft project report, Conduct peer review**
	1. Structure a Professional Project Report Prepare a well-organized draft project report covering problem definition, objectives, methodology, design, implementation, testing, and results. 2. Document Technical and Analytical Work Clearly present system architecture, algorithms, design decisions, and implementation details using appropriate technical language and formats. 3. Follow prescribed guidelines for formatting, referencing, plagiarism avoidance, and ethical reporting. 4. Communicate Project Outcomes Effectively Articulate project objectives, outcomes, limitations, and future scope in a clear and concise manner. 5. Conduct Constructive Peer Review Critically evaluate peers' project reports to identify strengths, gaps, inconsistencies, and areas for improvement.
WEEK- 12	**Finalize coding & documentation, Prepare demo/presentation slides**
	1. Finalize and Stabilize the Codebase Complete coding activities by resolving pending issues, optimizing logic, and ensuring code stability and correctness. 2. Apply Coding Standards and Best Practices Refactor code to improve readability, maintainability, and compliance with standard coding conventions. 3. Prepare Complete Technical Documentation Develop clear and comprehensive documentation including system overview, module descriptions, code comments, and user guides. 4. Ensure Traceability and Consistency Align finalized code with requirements, design specifications, and documentation for consistency and traceability. 5. Design Effective Demo Content Plan and organize a structured system demonstration highlighting key features, workflows, and innovations.

V. TEXTBOOKS:

1. K.N. Krishnaswamy, S. K. Dutt, “Engineering Design Process”, Pearson Education, India.
   Covers problem identification, requirement analysis, design methodology, and project execution.

2. Karl T. Ulrich, Steven D. Eppinger, “Product Design and Development”, McGraw-Hill Education.
   Focuses on concept generation, feasibility analysis, system design, and product development lifecycle.

3. Ian Sommerville, “Software Engineering”, Pearson Education.
   Useful for system architecture, UML, software development models, testing, and documentation.

4. Roger S. Pressman, Bruce R. Maxim, “Software Engineering: A Practitioner’s Approach”, McGraw-Hill Education.
   Provides practical guidance on requirements, design, coding, testing, and project management.

5. Harold R. Kerzner, “Project Management: A Systems Approach to Planning, Scheduling, and Controlling”,
   Wiley.
   Covers project planning, milestones, risk analysis, and execution strategies.

V. REFERENCE BOOKS:

1. David L. Olson, Desheng Dash Wu, “Enterprise Risk Management”, World Scientific Publishing.
   Useful for feasibility analysis, risk assessment, and decision-making.

2. Craig Larman, “Applying UML and Patterns”, Pearson Education.
   Strong reference for UML diagrams, system architecture, and object-oriented design.

3. Robert C. Martin, “Clean Code: A Handbook of Agile Software Craftsmanship”, Pearson Education.
   Supports best practices in coding, refactoring, and maintainable software development.

4. Glenford J. Myers, Corey Sandler, Tom Badgett, “The Art of Software Testing”, Wiley.
   Reference for unit testing, system testing, and validation techniques.

5. Donald A. Norman, “The Design of Everyday Things”, Basic Books.
   Useful for user-centered design, usability, and interface improvement.

6. Ramesh Venkataraman, Jeffrey K. Pinto, “Cost and Value Management in Projects”, Wiley.
   Supports economic feasibility and cost-benefit analysis.